Week 8: Lab

Module 4: Techniques

Collaboration level 0 (no restrictions). Open notes.

1. **Rod Cutting:** One of the examples discussed in detail in the week's lecture was the rod cutting problem. Download the notebook *RodCutting.ipynb*, and test your understanding of this problem by running through it step by step and reflecting on the answers. Fill in the missing function.

2. **Fibonacci numbers:** Consider the Fibonacci numbers, defined by

$$F(0) = F(1) = 1$$

and

$$F(n) = F(n-1) + F(n-2)$$

For example, the first several Fibonacci numbers are

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...$$

Consider the following recursive algorithm to compute the nth Fibonacci number:

```
Fibonacci(n)
    if n==0 or n==1:   return 1
    return Fibonacci(n-1) + Fibonacci(n-2)
```

(a) Is this algorithm correct?

(b) What is the running time of this algorithm? Is it $O(n)$? $O(n^2)$? $O(n^3)$?

(c) How would you make this algorithm better?

(d) Download the notebook *Fibonacci.ipynb* and run though it. Test your understanding of the problem by filling in the missing functions.

3. **Pharmacist problem:** A pharmacist has W pills and n empty bottles. Bottle i can hold p_i pills and has an associated cost c_i. Given W, $\{p_1, p_2, ..., p_n\}$ and $\{c_1, c_2, ..., c_n\}$, you want to store all pills using a set of bottles in such a way that the total cost of the bottles is minimized. So the problem is to find the minimum cost for storing the pills, and what bottles to use.

(a) Explain how the problem has optimal substructure.

(b) Give a recursive formulation of this problem (either as a function, or as pseudocode); if your function has arguments, make sure to explain what the arguments represent, and what your function will return.

Hint: Let $MinPill(i, j)$ *be the minimum cost obtainable when storing* j *pills using bottles among* 1 *through* i. *Thinking of the* 0-1 KNAPSACK PROBLEM *formulation may help.*

(c) Describe a top-down recursive dynamic programming algorithm with memoization and analyze its running time.

(d) Describe how to augment the algorithm in order to find the bottles used in the optimal solution.

(e) Describe and give pseudocode for a non-recursive dynamic programming algorithm for computing the optimal cost. Analyze its running time.

(f) Assume $n = 3, W = 7$, and $c[1] = 3; c[2] = 5; c[3] = 7$, and $p[1] = p[2] = p[3] = 1$. Call MinPill(3, 2) and show what entries in the table are filled, and their values.